# IMPLEMENTATION OF THE HIGH LEVEL ARCHITECTURE INTO DIS-BASED LEGACY SIMULATIONS

**Wesley K. Braudaway, Ph.D**
**Science Applications International Corporation**
**3045 Technology Parkway**
**Orlando, FL 32826**
**and**
**Susan M. Harkrider**
**USA STRICOM**
**12350 Research Parkway**
**Orlando, FL 32826**

KEYWORDS
High Level Architecture; Prototype Federation; DIS Legacy ReEngineering

ABSTRACT
The High Level Architecture (HLA) is a project to develop a simulation infrastructure to promote interoperability between simulations. The Defense Modeling and Simulation Office (DMSO) commissioned several experimental applications of HLA in 1996 to test and refine the HLA concept. As one of these experiments the Semi-Automated Forces (SAF) subsystems of the Close Combat Tactical Trainer (CCTT) were modified as a case study within a federation of three other DIS, platform-level simulation systems. In this paper, we describe the modifications made to CCTT SAF in order to incorporate the HLA into the CCTT SAF infrastructure.

## 1.0 BACKGROUND

The High Level Architecture (HLA) is a DoD-wide effort sponsored by the Defense Modeling and Simulation Office (DMSO) to aid in the establishment of a common technical framework. This framework facilitates the interoperability of all types of models and simulations, and facilitates the reuse of Modeling and Simulation (M&S) components. Initial definition of the M&S HLA was accomplished under the sponsorship of the Advanced Research Projects Agency's (ARPA) Advanced Distributed Simulation (ADS) program. It was transitioned to the DMSO in March 1995 for further development by the Architecture Management Group (AMG). For the effort, a set of prototypes were developed to address critical issues of the HLA. Each prototype, or federate, represented a major program, with both Government and industry technical teams. The experience of these prototypes helped to establish the HLA baseline, which was completed on 21 August 1996. The Under Secretary of Defense for Acquisition and Technology (USD(A&T)) approved the baseline as the standard technical architecture for all DoD simulations on 10 September 1996. The HLA, as a part of the common technical framework, is defined by three major components: the rules or functional definition; the object model template specification; and the interface specification. [1][2][3]

At this point, it may be useful to define several terms used by the HLA community, such as federate, federation, Runtime Infrastructure, simulation object model, and federation object model. A *federate* is a member of a HLA federation. A federate may include federate managers, data collectors, live entity surrogate simulations, model simulations or passive viewers. A *federation* is a named set of interacting federates, a common federation object model, and supporting Runtime Infrastructure, that are used as a whole to achieve a specific objective. The *Runtime Infrastructure* (RTI) is the general purpose distributed operating system software which provides the common interface services during the runtime of an HLA federation. To facilitate the formation of a federation, each federate is required to develop a *Simulation Object Model* (SOM). A SOM is a specification of the intrinsic capabilities that an individual simulation publicly offers to federations. The standard format in which SOMs are expressed provides a means for federation developers to determine the suitability of simulation systems to assume specific roles within a federation. Once the federation has been formed, the *Federation Object Model* (FOM) is developed. A FOM is an identification of the essential classes of objects, object attributes, and object interactions that are supported by a HLA federation.

This paper describes defines compliance as it exists today, defines the process of implementing HLA in DIS systems, and then provides an assessment of that integration. The paper also addresses open issues for DIS as part of the HLA, and capabilities that will be available in the near future.

## 2.0 COMPLIANCE

Compliance is currently defined as how well a federate handles the functions defined in the HLA interface specification, whether it has an object model in conformance with the Object Model Template (OMT), and whether it complies with the HLA rules for federates. The interface specification functions are as described in paragraph 3.1.2 of this paper. The OMT provides the format to specify the objects intrinsic to a given federate or federation. The HLA rules separate the regulations between those for federates and those for federations, with an equal number of rules for each.

Federation rules can be summarized to state that in order to achieve compliance:

- Federations shall have a HLA FOM documented in accordance with the OMT;
- Representation of objects in the FOM shall be in federates, not the RTI;
- Exchange of FOM data shall occur via the RTI during federation execution;
- Federates shall interact with the RTI in accordance with the HLA interface specification during federation execution; and,
- During a federation execution, an attribute of an object's instance shall be owned by only one federate at any given time.

Similarly, rules for federates can be summarized:

- Federates shall have a HLA SOM, documented in accordance with the OMT;
- Federates shall be able to update and/or reflect any object attributes in their SOM, and send and/or receive SOM interactions externally;
- Federates shall be able to transfer and/or accept ownership of attributes dynamically during federation execution, as specified by a federate's SOM;
- Federates shall be able to vary the conditions under which updates of object attributes are provided; and,

- Federates shall be able to manage local time such that they can coordinate data exchange with other members of a federation.

For clarity, compliance is different than federation testing, which is the responsibility of individual federations. The HLA was tested prior to the baseline definition through the prototypes. The prototypes provided a variety of conditions and scenarios which addressed a range of potential DoD M&S applications. Another dimension of testing, performance testing, took place in the proto-federations using a common performance measurement framework and a common interface test procedure. The feedback from the proto-federations was used to revise the existing test procedures as a supporting document to the HLA baseline.

## 3.0 IMPLEMENTING HLA COMPLIANCE

To re-engineer a legacy DIS system into a HLA compliant federate requires developers to define a SOM, to integrate the RTI Interface and HLA functionality into the system, and transform the system's DIS oriented data model into the new SOM data representation. This section describes the integration tasks and describes the re-engineering case study for the SAF subsystem of the CCTT system.

### 3.1 HLA Integration Tasks

The major re-engineering design decisions can be characterized along three dimensions: the degree to which the system's capabilities are to be shared with other federation members, the degree to which the system will support and utilize the HLA services, and the degree to which the system's data model is modified to match its new SOM. The first dimension affects the definition of the system object model and depends strongly on the system's role in a range of federations. The second dimension determines the amount of modification required to support previously unsupported system capabilities that are now provided using the HLA (for example, simulation modeling ownership transfer). The third dimension affects the extent to which the legacy system's data flows and data model are modified to achieve HLA compliance. Each of these dimensions are discussed further in this section.

#### 3.1.1 System and Federation Object Model

The SOM development is the first design step of re-engineering a system to become HLA compliant. The more completely the SOM describes the system's total capacity for publicly representing its objects, attributes, and interaction, the more system re-engineering effort will be required to implement HLA compliance. However, only those system capabilities represented within the SOM can be shared with other federates in a HLA compliant implementation. The SOM is used as critical information for determining the composition of federations and the specification of a federation object model (FOM) that specifies all public data shared among the federates. Some re-engineering effort might be saved if after considering the range of federations in which the system may participate, the full set of system capabilities are not required. In this case, transforming the system's complete communication capabilities will not be necessary.

Therefore, to design the SOM, both the role of the system in a range of federations and a set of typical scenarios for the federations can be helpful for designing the objects, attributes, interactions, and transmission policies of the SOM. The SOM (and the FOM for a federation) are represented using the format defined by the HLA Object Model Template [3].

As more DIS legacy systems are re-engineered for HLA compliance, a set of "DIS reference FOMs" or DIS FOM standards will emerge that will reduce the re-engineering effort and maximize the reuse of modified DIS systems in many platform oriented federations. The need for DIS FOM standardization stems from the flexibility provided by the HLA OMT for defining SOMs and FOMs. To transform the DIS protocol into an object model several basic design decisions become apparent: What is the object hierarchy for a DIS FOM? What are the set of important attributes for each object? What are the interactions? One option, for example, is to define a FOM object hierarchy that matches the DIS enumeration hierarchy. While this will produce a large number of object classes (one for each type of simulated entity) it maximizes the use of HLA's class-based filtering. A second approach is to encode the Entity State Protocol Data Unit (PDU) as a FOM class where its attributes correspond to the attributes of the Entity State PDU. Since the options are numerous, a proliferation of individual DIS like federations will emerge with little chance of being interoperable without a reference DIS FOM.

#### 3.1.2 Interface Functionality

The second major task of re-engineering a system for HLA compliance is to map the HLA interface services to the supportable capabilities of the system. The RTI implements the HLA interface which provides support for federation management, declaration management, object management, ownership management, time management, and data distribution management services. To redesign a system as an HLA federate, the system will not only interface to and use services of the RTI but also provide services (Federate Services) that respond to RTI's callbacks. The extent to which a system can implement or can be extended to implement the HLA functionality depends on its capabilities and its roles in a set of anticipated federations.

**Federation Management Service** This service includes Create, Join, Pause/Resume, Save/Restore actions on a Federation Execution. The extent to which this service corresponds to existing system capabilities will influence the effort necessary to re-engineer these capabilities.

**Declaration Management Service** This service includes Publish, Subscribe, and Control actions on specific classes and interactions. DIS systems expect to send and receive all public information available and have never specifically requested particular classes of information, specified the classes they will transmit, or accepted control specifying when to start or stop individual object transmissions. These extended capabilities will be required for any HLA re-engineered effort.

**Object Management Service** This service includes Requesting ID numbers, Updating object attributes, Sending interactions, Receiving object updates, Receiving interactions, Deleting objects, and Changing transportation characteristics. These capabilities are new to DIS legacy systems other than the transmission capability of sending a complete entity state PDU with a minimum update rate. For re-engineering, the new options include sending only changed attributes, sending only a subset of attributes at a minimum rate, and changing the characteristics of the transmission (i.e., reliable, best effort, receive order, time stamp ordered, etc.).

Another important issue for DIS systems is the assignment of object IDs. DIS applications are responsible for generating their own IDs and usually attach some semantics to the ID such as a site number, application number, and entity number within the application. For HLA the ID is generated by the RTI and is meaningless to the applications except to uniquely identify an object. If the DIS system depends on the semantics associated with the DIS entity ID then this information must be explicitly maintained within the designed FOM.

**Ownership Management Service** This service implements the transfer of simulation responsibility for individual object attributes between federates. Although DIS applications have occasionally had a requirement to transfer entity ownership, the DIS standard has not effectively supported this requirement and most systems have either removed the requirement or have used some non-standard method for implementation. In either case, to utilize this service will require considerable re-engineering effort.

**Time Management Service** DIS legacy systems typically run in real-time and have their own means for synchronizing and managing the passage of simulation time. The current HLA time management service provides support for time-step and event-driven simulation systems but offer no standard time management support for platform level real-time simulations.

**Data Distribution Management Service** This service provides federate control over the routing of attributes based on the attributes' data values. This is a new capability to DIS legacy since DIS systems typically use a data broadcast approach. Recent DIS developments, however, have utilized multicast network capabilities to help filter DIS traffic at least by exercise number if not also by geographic characteristics of the data. This HLA service provides a means for implementing these multicast capabilities and may provide more extensive data distribution management than previously available to DIS legacy systems.

### 3.1.3 Data Model Translation

The internal data model implemented by a DIS legacy system will likely differ substantially from the resulting SOM that will be implemented to make the system HLA compliant. Some means of transforming the system's data model into the SOM data model must be conducted as part of the re-engineering effort. This is the third major task in re-engineering a DIS system to HLA compliance. Since the DIS system was designed to communicate using the DIS protocol syntax externally, much of its internal data model will be oriented toward this DIS protocol syntax. To communicate with other federates using objects, attributes, and interactions defined by a FOM, some transformation of this internal data model must be conducted as part of the re-engineering task. Alternatively some run-time translation must be implemented in the data flow between the DIS simulation application and the RTI interface. The next section describes these alternatives in more detail.

### 3.2 Implementation Methods and Implications

Several design options are available for transforming an existing DIS compliant system into a HLA compliant system. The approaches can be characterized as either a gateway approach or some level of object model integration. Figure 1 shows three levels of HLA integration: a gateway approach, a transport layer integration, and a complete HLA integration. In each approach the data model transformation occurs at some phase of the re-engineering development

In the first approach, Gateway, the federate communicates through its network connection using the DIS protocol. A DIS Converter captures this DIS communication from the network and transforms it into an Object/Attribute Database for communication using the HLA services provided by the RTI. The data model translation occurs during run-time in the gateway process. Although this is the simplest approach for implementing HLA compliance with respect to the impact on the legacy system, the throughput of the resulting DIS Converter is likely to be lower than can be tolerated. Additionally, the application's use of the HLA services are limited to those that can be represented directly using DIS, thereby limiting the advantages of becoming HLA compliant.

The second alternative, Transport Layer Integration, integrates an RTI interface module into the transport layer of the application. In this approach the data model translation to the FOM is conducted within the transport layer before leaving the system's run-time environment. The simulation application of the system will continue to use its DIS oriented data structures as they flow to the transport layer where they are translated into FOM values. Using this approach there is little or no impact to the DIS application assuming that the system architecture isolates the simulation application from the transportation mechanisms. While this alternative removes the latency issues of the gateway approach and requires minimal re-engineering effort, it does incur some latency by translating the application's internal DIS-oriented data model into FOM objects and interactions.

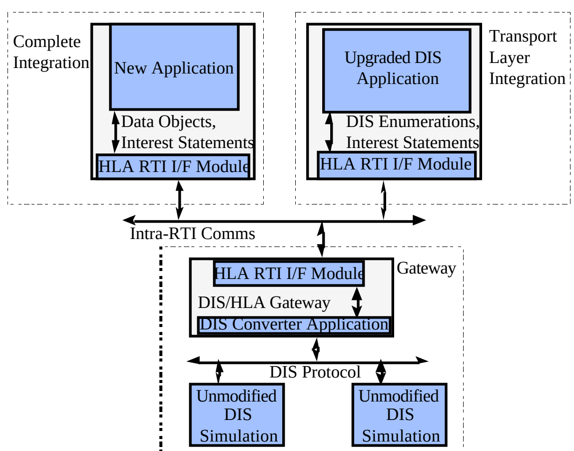


Figure 1. DIS Legacy Integration Approaches

The third alternative is a complete integration of HLA into the simulation application. Through this approach, the data models within the application would be transformed into objects and behaviors represented by the system's SOM. The internal data would not be influenced by the DIS protocol representation but would be made more general in terms of communicated objects, attributes and interactions. Hence the data model transformation is conducted during the re-engineering rather than as a function of the run-time implementation.

### 3.3 CCTT SAF Case study

The STRICOM/PM CATT Close Combat Tactical Trainer (CCTT) is the first fully Distributed Interactive Simulation (DIS) compliant training system that will train Armor, Cavalry, and Mechanized Infantry platoons through Battalion/Task Forces on their doctrinal Mission Training Plan collective tasks. The system consists of networked vehicle simulator manned-modules in combination with Semi-Automated Forces (SAF), and other workstation systems. The SAF system has the capacity to create a wide variety of friendly or opposing force units and vehicles that exhibit highly realistic behaviors to support the CCTT training objectives.

CCTT SAF was modified as a case study within a federation of three other DIS, platform-level simulation systems to demonstrate and evaluate the HLA approach. This case study also served to expose the issues involved in re-engineering a DIS application for HLA compliance.

### 3.3.1 Scope of the Federation Object Model

The FOM developed for this DIS study was based upon the sole objective of using the current capabilities of the four platform-oriented simulation systems to test the concepts of the HLA. With this objective, a scenario was developed that represented a somewhat realistic battlefield occurrence, used the predominate capabilities of each federate, and tested some of the key HLA capabilities (e.g., ownership transfer, class/attribute type filtering). The FOM was specifically developed to support this one scenario and represented a small subset of the SOMs provided by each federate in the study.

As a DIS-based federation, the FOM (especially its earlier versions) represented a small portion of the data model defined by the DIS 2.0.4 standard. The object class structure was essentially equivalent to the DIS Entity enumeration for those entities identified in the scenario. The attribute table applied to all entities and included all attributes as specified by the Entity State PDU. The Collision, Detonation, and Fire PDUs were translated into FOM interactions in this FOM. The federation made a strong assumption that data values and semantics of object updates and interactions were documented by the DIS 2.0.4 standard and were not discussed or documented as part of the FOM development.

For this case study, the FOM did not include nor test the implications of Simulation Management, Event Reporting, the SAF command and control protocol, the Synthetic Environments protocol, radio transmissions, re-supply, repair, and other critical capabilities of CCTT. Because some of these capabilities are critical to operation of the CCTT SAF system, some existing DIS communication capability was retained by the prototype CCTT SAF design described in this report. A complete re-engineering for HLA compliance would not require this DIS communication flow.

### 3.3.2 CCTT Architecture

Each simulation application within the current CCTT baseline has an architecture as shown in Figure 2. Some number of simulation or workstation application processes on a single processor access an entity database and PDU input/output queues contained in shared memory. A separate Entity Processor process maintains the dynamic state of the entity database and the DIS Network Manager process manages the flow of DIS PDUs to and from a FDDI Network. The Entity Processor maintains the dead reckoning of relevant remote entities and determines whether local entity information should be transmitted to other simulations To minimize latency, the DIS Network Manager places incoming DIS Entity State PDUs into a holding area in the entity database, where they are subsequently processed by the Entity Process to make the data available to applications. Likewise, for each local entity update that are to be transmitted, the Entity Process places these entities by reference into the PDU Queue for processing by the DIS Network Manager. The DIS Network Manager uses this reference to directly access the Entity State PDU in the Entity Database and send it to the FDDI Network. This method is facilitated by the DIS Entity State PDU format of each entity in the entity database; therefore, the Entity Processor and the DIS Network Manager operate on the same DIS data model.

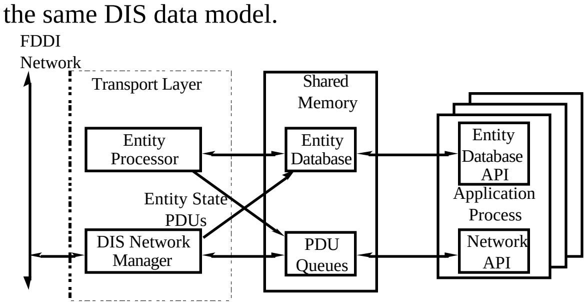


Figure 2. CCTT Baseline Architecture

### 3.3.3 System Architecture Design Summary

For this HLA case study, the Entity Processor and the DIS Network Manager of CCTT's transport layer were modified to communicate FOM data using a prototype RTI as shown shaded in Figure 3. This approach corresponds to the Transport Layer Integration approach described in Section 3.2. The Network Manager translates certain outgoing DIS PDUs (Fire, Collision, and Detonate PDUs) into FOM interactions and issues these interactions using the RTI. Conversely, the Network Manager receives interactions from the RTI, translates them into corresponding DIS PDUs, and processes them as if they were received from the FDDI network. For this study, existing data flows for DIS communication were maintained for incoming DIS traffic that was not included in the FOM (i.e., simulation management and SAF command and control protocol).

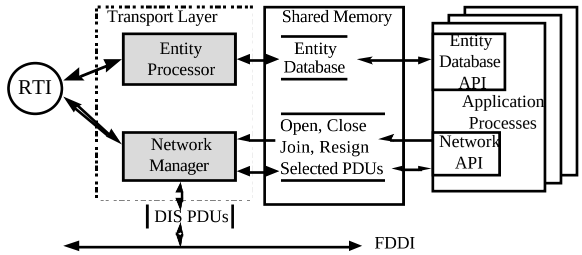


Figure 3. CCTT RTI Integrated Architecture

The modified Entity Processor collects, translates, and communicates object updates to and from the RTI. For each locally simulated entity, this process monitors the entity updates in the Entity Database, collects those attribute values that change beyond their dead reckoning thresholds, and sends the entity's attributes as an RTI object update. For remote entity object updates, this process receives the update and modifies the remote entity's attributes in the Entity Database.

Through this design each CCTT processor, containing some number of simulation applications, acts as two federates to the RTI regardless of the number or type of applications running on that box. The Network Manager is a federate concerned exclusively with RTI interactions while the Entity Processor is a federate concerned exclusively with objects.

Minor modifications were made to the applications to request that the Network Manager and Entity Processor join and resign the federation execution. A complete integration within CCTT would also require that each application specify the objects and interactions that they can publish and subscribe.

## 4.0 FOM DEVELOPMENT ASSESSMENT

The FOM development and resulting performance of the federation test for this DIS study raised issues with respect to the FOM development methodology and federate coordination. Specific lessons were learned about the FOM development:

- Deliberate tradeoffs must be made between the DIS Entity State approach to specifying entity objects and a pure object-oriented approach.
- A complete specification of syntactic, semantic, and behavioral requirements must be fully specified by the FOM in order to maximize the chance of interoperability.

The initially developed FOM defined an object-oriented component structure for articulated parts that differed from the typical DIS approach of specifying articulated part attributes for particular entities. For example, the FOM defined tank, turret, and main gun object classes. During the creation of a tank, the simulation application would create an instance of the tank, the turret, the main gun, and through parent/child attributes of each class, linked the objects together. This approach added a level of complexity to each application since the HLA does not provide services to help create or manage these object associations. For example, if an application receives a new occurrence of a main gun, it must hold onto this partially specified entity until all the pieces (i.e., tank and turret) have arrived; otherwise, the application risks never receiving the main gun information again. The result is that objects are fragmented and each application is responsible for piecing the fragmented object together without aid of any HLA object component services. To remove this complexity, the FOM was redesigned to include the articulated part attributes to those classes where turrets and main guns where relevant.

A more fundamental issue with the FOM development did not appear until late into the federation testing phase of the study. As stated earlier there was an assumption that much of the semantics specifying the conditions under which data is transmitted and actions in response to received data were understood as specified by the DIS 2.0.4 standard. Therefore, the behavioral implications of interactions and attribute modifications were not discussed nor recorded in the FOM. Several instances occurred during federation testing that exposed this invalid assumption. One such example occurred when one federate's fixed wing air dropped a laser guided bomb that destroyed a CCTT SAF tank. Rather than relying upon an attribute update for the tank identifying it as destroyed, the other federate expected CCTT to delete the vehicle from the environment. As a result, subsequent bombing runs bombed only the already destroyed tank even though there were other undamaged tanks nearby.

Other behavioral and semantic inconsistencies arose during the federation test. The existence of a FOM alone does not guarantee interoperability as shown by these federation tests. The study demonstrated that the HLA object model template should be extended to include more semantic and behavioral specifications to ensure that all members of the federation understand their data model requirements.

**5.0 LEVEL OF EFFORT**

The effort involved in re-engineering a DIS legacy simulation depends on the three dimensions described previously: how much data is made public to the federation, how much re-engineered and new functionality is implemented using the HLA services, and the degree to which the system's data model is modified to match its SOM.

It is estimated that the re-engineering of the entire CCTT system using the transport level implementation approach developed during this study will require, a rough order of magnitude, 30,000 source lines of Ada95 code. For this study the CCTT SAF re-engineering developed 5,000 source lines of mainly Ada95 code development. This estimate does not account for implementing any additional HLA functionality that is currently not supported by CCTT such as the Ownership Management Services.

**6.0 OPEN ISSUES FOR DIS**

The HLA development has only reached its first phase and will evolve vastly over the next few years. From the perspective of DIS platform-level legacy systems several issues remain unresolved or in progress. These issues include real-time management, geographic filtering, HLA scalability, and FOM partitioning.

**6.1 Real-time Management** The ability of federates to exchange temporal information during the real-time simulation is critical for managing the affects of highly latent networks. For example, this timing information can be used to modify position data received late by dead reckoning the data to be consistent with the current real-time. The RTI time management services do not support the coordination of these real-time causal relationships. Real-time platform level simulations rely on the synchronization of their system clocks and the faithfulness of each system to the correspondence between simulation event timings and real-time as represented by the system clock. To ensure an accurate report of timed events for those simulations that required this information, the case study federation chose to use the Network Time Protocol to synchronize each federate's system clock.

**6.2 Geographic Filtering** HLA's data distribution management should be evaluated as a replacement for the current DIS geographic or "Area of Interest" filtering schemes. The capabilities of this service appear to meet the needs of CCTT although its performance is unknown. During the reported study, this HLA service was not available. The correct use of this capability is a subject of further study.

**6.3 HLA Scalability** The case study demonstrated scalability concerns due mainly to the use of a prototype RTI whose latency did not permit experimenting with large scenarios. A "large" CCTT exercise would consist of several hundred entities all interacting in a federation execution. The latency problems that arose during the case study prevented any testing that would validate the RTI's capabilities in this area. Studies are proceeding to ensure that the HLA concept of a central distribution agent, the RTI, can be implemented so as to support real-time federation executions with thousands of participating entities.

**6.4 FOM Partitioning** The CCTT SAF development also demonstrated the need for federates to be able to simultaneously join multiple federation executions using the RTI. This capability will allow federates to share their public data within multiple federations concurrently where each federation satisfies a specific need of the federate. For example, CCTT would define a separate federation execution for a site's Simulation Management, for each concurrent exercise, and for each exercise's SAF command and control communication. For a specific exercise, this implementation will allow the CCTT SAF system to behave as a single federate to the federation execution implementing an exercise. The SAF system, however, would be internally distributed (i.e., a distributed federate) and would use a command and control FOM within its distributed HLA compliant implementation.

**7.0 REFERENCES**

[1] HLA Interface Specification, Defense Modeling and Simulation Office, Version 1.0, 15 August 1996.
[2] HLA Rules, Defense Modeling and Simulation Office, Version 1.0, 15 August 1996.
[3] HLA Object Model Template, Defense Modeling and Simulation Office, Version 1.0, 15 August 1996.